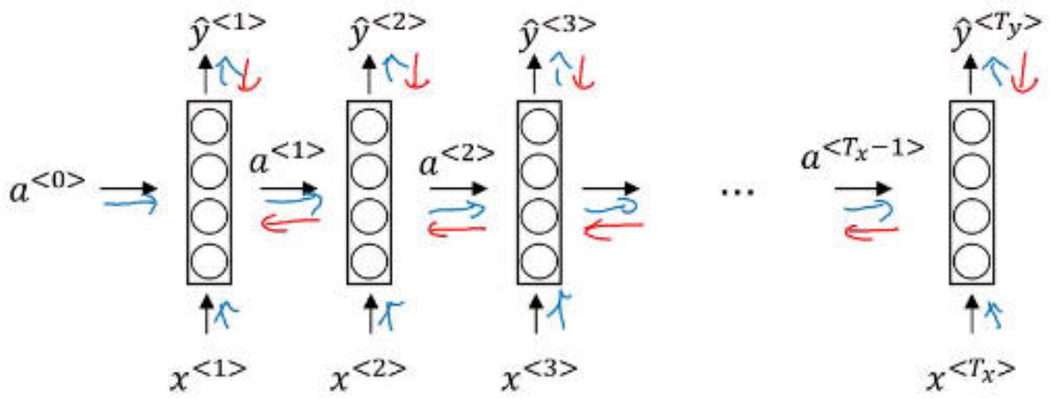
1 14

循环神经网络(RNN)

为什么不用NN做NLP? { 输入输出维度不定
参数庞大(One hot if)
特征不共享.

Forward propagation and backpropagation



Andrew Ng

数学表示: $a^{<1>} = g(W_{ax}x^{<1>} + W_{aa}a^{<0>} + b_a)$ tanh/Relu $a^{<0>} = \vec{0}$.

$\hat{y}^{<1>} = g(W_{ya}a^{<1>} + b_y)$ → sigmoid/SoftMax.

$a^{<t>} = g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$

$y^{<t>} = g(W_{ya}a^{<t>} + b_y)$

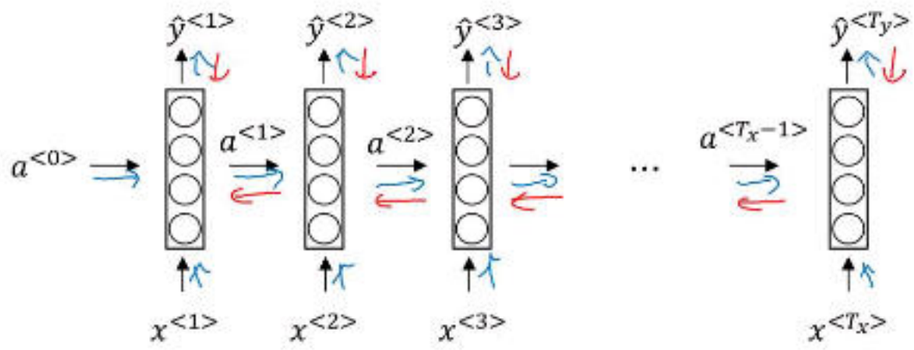
化简: $a^{<t>} = g(W_a\begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix} + b_a)$

$y^{<t>} = g(W_y\begin{bmatrix} a^{t} \end{bmatrix} + b_y)$.

1 14

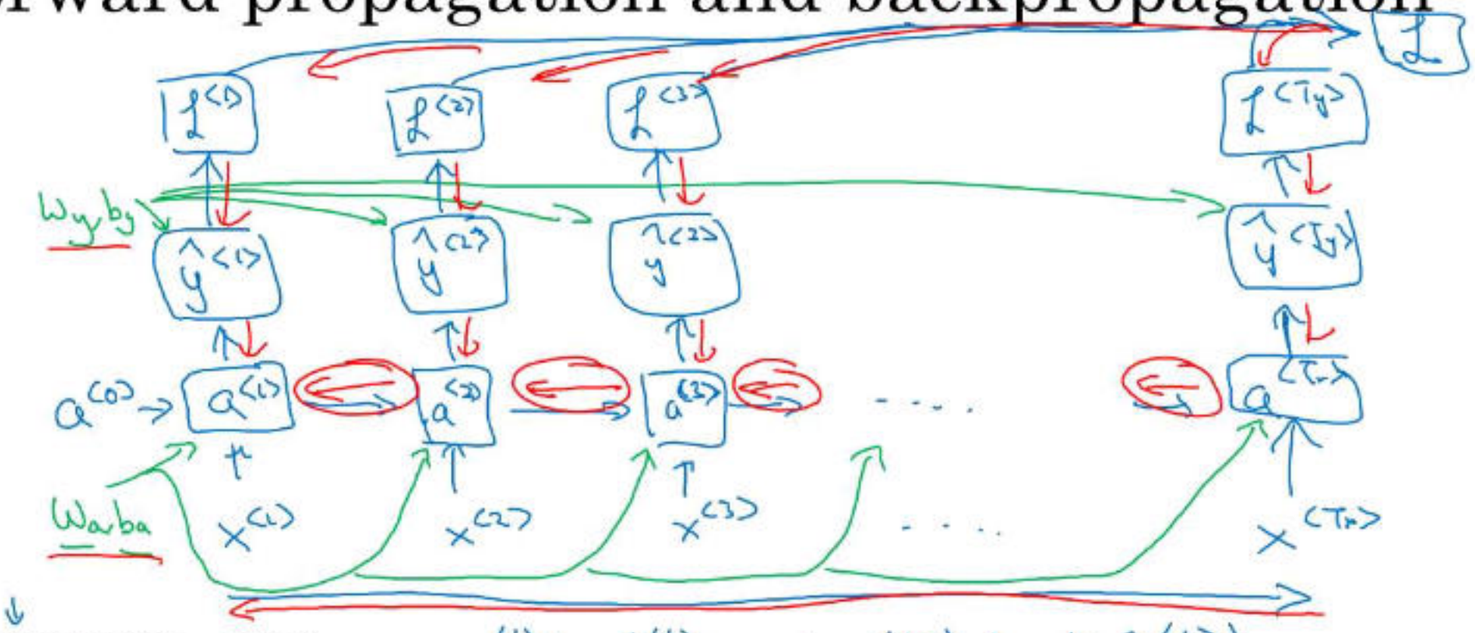
Back Propagation.

Forward propagation and backpropagation



Andrew Ng

Forward propagation and backpropagation



$$\mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>}) = -y^{<t>}\log\hat{y}^{<t>} - (1-y^{<t>})\log(1-\hat{y}^{<t>})$$

$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y}\mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>})$$

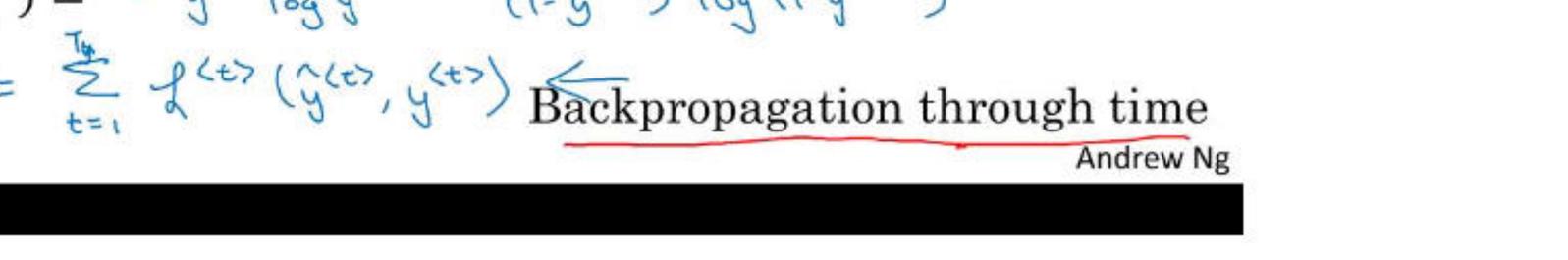
Backpropagation through time

Andrew Ng

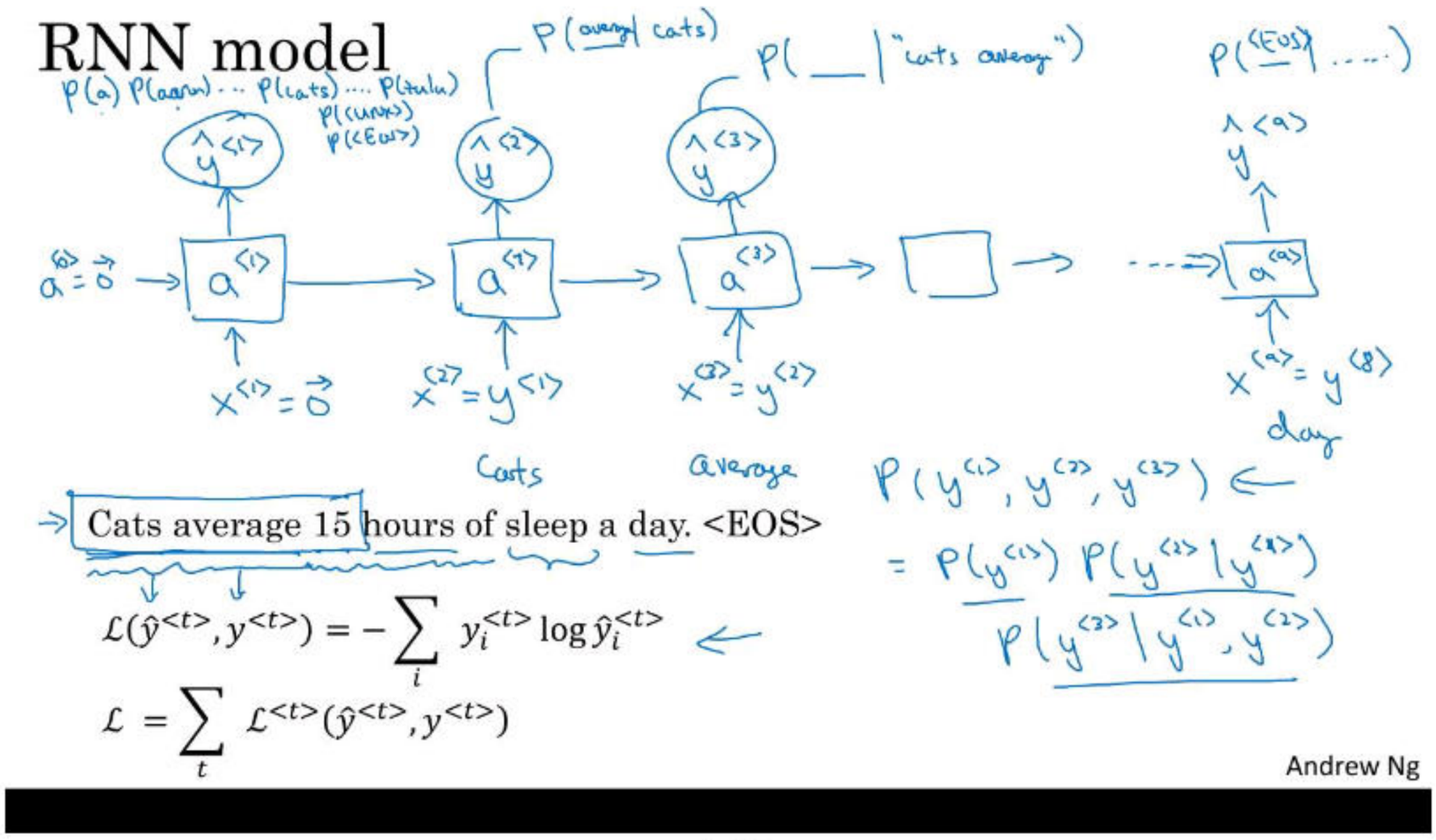
损失函数可采用交叉熵.

Summary of RNN types

各种 RNN:



Andrew Ng



Language Model: $P(\hat{y}^{<t>} | y_1^{t-1})$. n元文法.

$\hat{y}^{<t>}$ 是 SoftMax, Loss 用交叉熵.

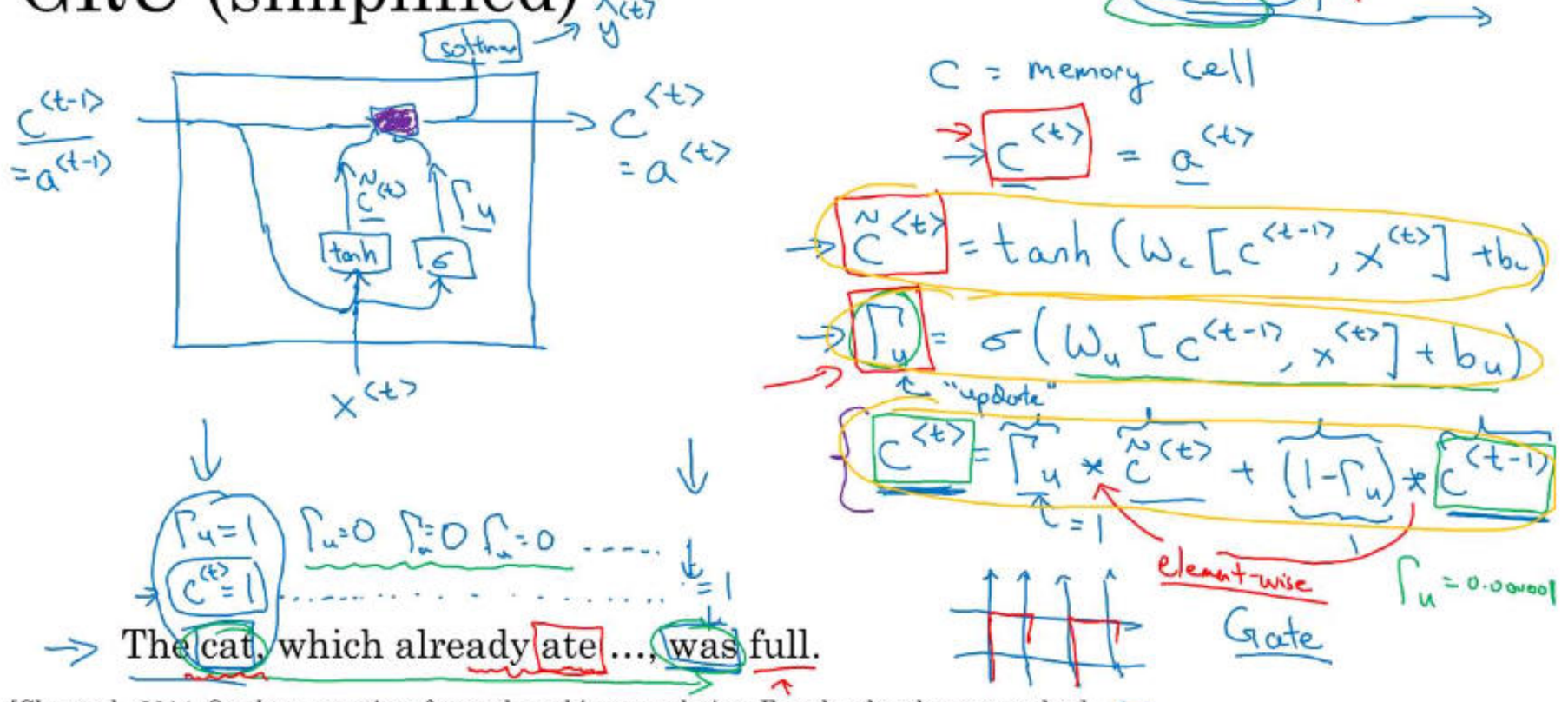
在训练好的语言模型上可以采样产生词序列. (文本生成)

Gradient Vanishing.

句子有一定长度, 容易梯度消失.

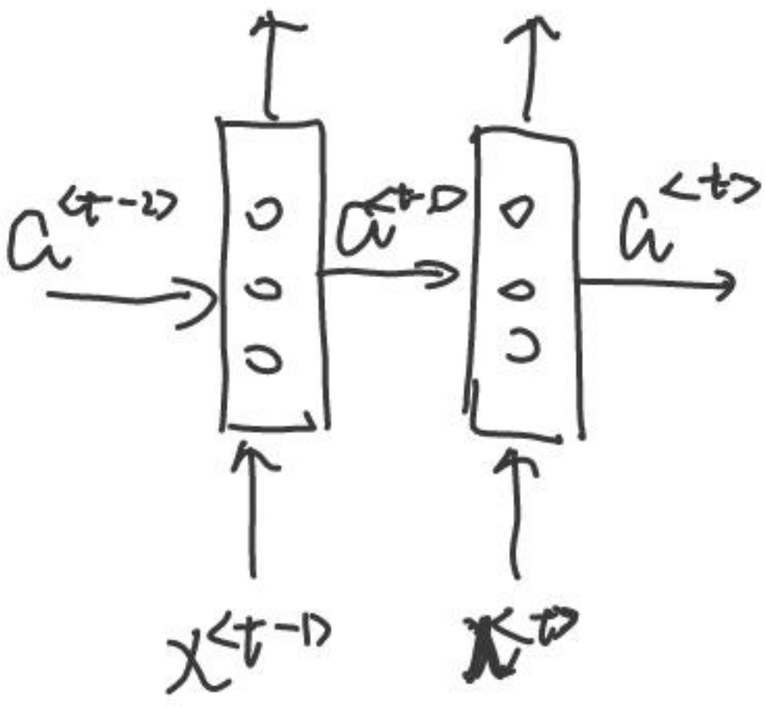
梯度爆炸可以通过 Gradient Clip 解决.

GRU (simplified)



[Cho et al., 2014. On the properties of neural machine translation: Encoder-decoder approaches]
[Chung et al., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling]

Andrew Ng



通过 $c^{\langle t \rangle}$ 记忆信息.

Cell 通过 Γ_u 判断是否更新.

1 14

LSTM.

LSTM in pictures

$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

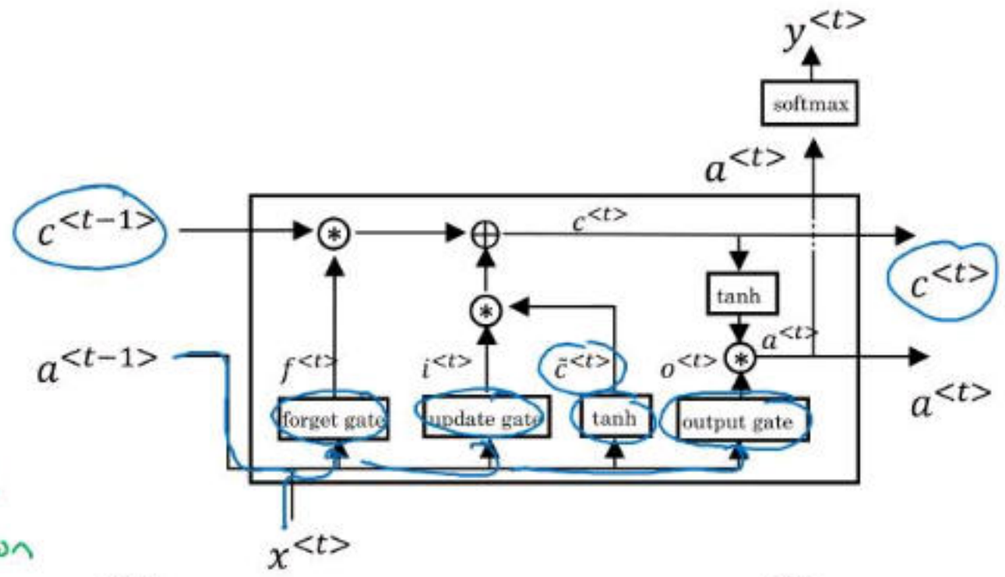
$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

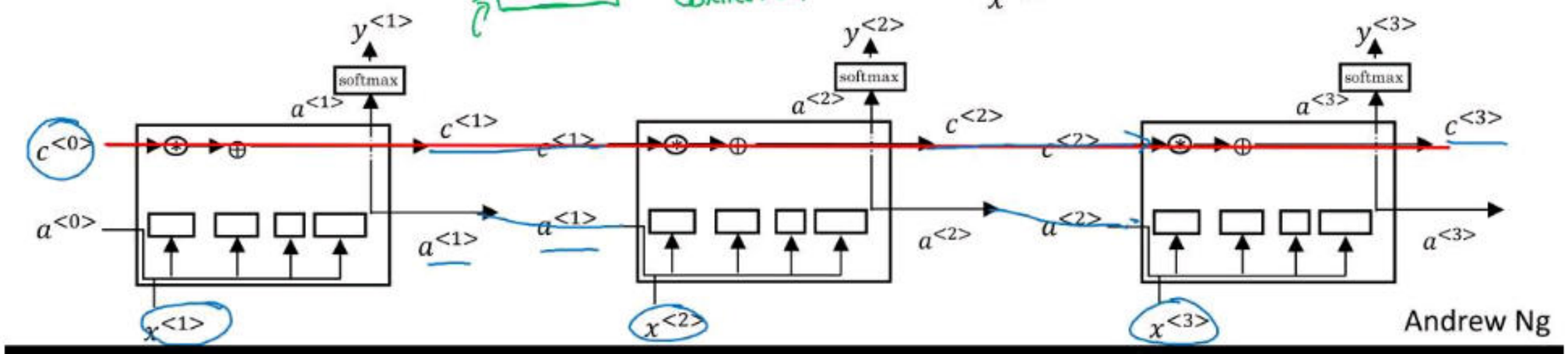
$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

$$a^{<t>} = \Gamma_o * c^{<t>}$$

$c^{<t-1>}$ peephole connection





引入 Γ_f，遗忘门、输出门 Γ_o，

Cell 控制更新多少信息，遗忘多少信息。